

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student: **Michael Libotovský**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: E LINKX a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Gaura**

Konzultant bakalářské práce: Ing. Roman Hrdý

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 6. dubna 2015

.....Libor Loucky.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. dubna 2015

.....Libor Loucky.....

Rád bych na tomto místě poděkoval všem, kteří mi s absolvováním individuální odborné praxe pomohli. Především bych chtěl poděkovat zaměstnancům společnosti E LINKX a.s. Ing. Romanu Hrdému, Ing. Radku Petříkovi a Ing. Martinu Gürtlerovi za přátelský přístup a jejich trpělivost v průběhu praxe.

Abstrakt

Tato práce popisuje průběh vykonávání bakalářského semináře formou individuální odborné praxe u společnosti E LINKX a.s. Popis absolvování této praxe je rozdělen na dvě samostatné části, kde první část je zaměřena na absolvování školení a druhá část je zaměřena na následnou stáž v provozu firmy. V této práci jsou shrnuty úkoly spolu se stručnými popisy řešení nebo náplní. Závěrem student rekapituluje své působení ve společnosti E LINKX a.s.

Klíčová slova: E LINKX a.s.,SQL,.NET

Abstract

This bachelor thesis describes the course of the individual professional practise in the E LINKX a.s. company. Bachelor thesis is divided into two separate parts, the first part is focused on completing the training, and the second part focuses on the subsequent internship in business operations. In this work are summarized tasks along with brief descriptions of the solutions or refills. Finally, student recapitulates his role at E LINKX as.

Keywords: E LINKX a.s.,SQL,.NET

Seznam použitých zkratek a symbolů

GUI	– Graphical User Interface
SQL	– Structured Query Language
T-SQL	– Transact-SQL
WCF	– Windows Communication Foundation
XML	– Extensible Markup Language
ASP.NET	– Active Server Pages
ASPX	– Active Server Page Extended File
RDP	– Remote Desktop Protocol
IIS	– Internet Information Service
ERP	– Enterprise resource planning
B2B	– Business-to-business
B2C	– Business-to-consumer
API	– Application Programming Interface
SOAP	– Simple Object Access Protocol
MSDN	– Microsoft Developer Network

Obsah

1	Úvod	6
2	Popis firmy	7
2.1	Produkty	7
2.1.1	HelpDesk	7
2.1.2	Esyco.NET	7
2.1.3	xForLink	7
2.1.4	xSklad	7
2.1.5	xShop	8
2.2	Pracovní prostory	8
3	Popis použitých technologií	9
3.1	C#	9
3.2	ASP.NET	9
3.3	T-SQL	9
4	Školení	10
4.1	Replikace dat	10
4.1.1	Tým, analýza částí a nacenění práce	11
4.1.2	Windows služba	11
4.1.3	Konfigurační aplikace	12
4.1.4	Notifikační aplikace	12
4.1.5	Implementace Windows služby	12
4.1.6	Implementace konfigurační aplikace	13
4.1.7	Implementace Notifikační aplikace	14
4.1.8	Rozšíření: Správa historie	14
4.1.9	Meziaplikační komunikace	14
4.2	Webové rozhraní pro replikaci dat	15
4.2.1	Tým a analýza zadání	15
4.2.2	Implementace aplikačního rozhraní	16
4.2.3	Implementace webu	16
4.3	Parametrizace XML	17
4.3.1	Použitý software a rozdělení práce	17
4.3.2	Průběh práce a její dokončení	18
5	Stáž	19
5.1	Oddělení FLEXICA	19
5.1.1	Testování a testovací scénáře	19
5.2	Oddělení Logistiky	19
5.2.1	Tvorba tabulek a číselníků	20
5.2.2	Vývoj produktu	20
5.2.3	Generování cenové nabídky	20

6	Závěr	22
6.1	Uplatnění teoretických a praktických znalostí ze studia	22
6.2	Scházející odborné dovednosti a znalosti	22
6.3	Shrnutí	22
7	Reference	23
	Přílohy	23
A	Hodinové ohodnocení replikace dat	24
B	Ukázka openXML dokumentu	26
C	Popis metody WdAddTable	27
D	Výčet dostupných formátů tabulky	28

Seznam tabulek

1	Tabulka ohodnocení: Konfigurační aplikace	24
2	Tabulka ohodnocení: Služba	24
3	Tabulka ohodnocení: Notifikační aplikace	25
4	Tabulka ohodnocení: Ostatní	25
5	Tabulka ohodnocení: Projekt replikace dat	25

Seznam obrázků

1	GUI: Replikator	14
2	GUI: Web	17

Seznam výpisů zdrojového kódu

1	Služba: Hlídání změn nad svým XML souborem	13
2	Ověření existence služby a získání její konfigurace	13
3	Vyhledávání zpráv v EventLogu	16
4	Přepnutí jazyka zobrazované stránky	17
5	Vyhledávání klíče v odstavcích	21
6	Ukázka dokumentu v openXML	26
7	Signatura metody WDAddTable	27
8	Výčet formátů kterými lze ovlivnit finální vzhled tabulky	28

1 Úvod

Tato bakalářská práce popisuje průběh absolvování odborné bakalářské praxe ve společnosti E LINKX a.s. Se společností jsem byl seznámen na prezentaci firem, kterou pořádala fakulta elektrotechniky a informatiky na konci čtvrtého semestru roku 2014. Společnost jsem osobně kontaktoval a po domluvě jsem byl pozván na přijímací pohovor do sídla společnosti v Ostravě-Mariánských Horách. V rámci přijímacího pohovoru jsem musel dokázat jak teoretickou znalost jazyka C# a teoretickou znalost databázových systémů, tak i praktickou dovednost jazyka C# implementací zadaných úkolů.

Odbornou praxi jsem začal vykonávat již v červenci 2014 a počátkem září 2014 jsem byl přijat do společnosti E LINKX a.s. na pozici stážisty. Z tohoto důvodu je celá tato práce rozdělena na dva celky, tedy na období před stáží, které jsem nazval "*školení*", a na období následné stáže.

V části školení popisuji plněné úkoly a projekty, které nám byly zadány, spolu s jejich postupem řešení. Druhá část je zaměřena na stáž, kde z důvodu citlivosti informací společnosti pouze nastiňuji zadávané úkoly a jejich řešení.

Tato práce obsahuje i sekci příloh, kde naleznete veškeré obrázky, tabulky a ukázky kódů, na které se v průběhu této práce odkazují, a jsou příliš rozsáhlé pro začlenění do textu. Ukázky kódů jsem si dovolil ve prospěch čitelnosti a z důvodu možného vytržení z kontextu zkrátit, a to především o bloky zachycování výjimek a ošetřování vstupů.

2 Popis firmy

Společnost E LINKX a.s. je ryze česká softwarová společnost. Byla založena roku 1999 a již od počátku patří k partnerům eD' system. V současné době se společnost E LINKX a.s. zabývá vývojem a implementací informačních systémů a webů pro obchody, sklady a logistiku [1].

2.1 Produkty

2.1.1 HelpDesk

Produkt zaměřený na správu interních procesů mezi zaměstnanci společnosti a komunikaci se zákazníkem. Tento produkt obsahuje základní výkaz práce, kde si zaměstnanec eviduje svou práci v průběhu dne (co dělal a jak dlouho to dělal) a komplexní systém požadavků (úkolů), které jsou mezi zaměstnance rozdělovány. Každý požadavek má svůj popis, přílohy a stavy (Schváleno, V řešení, Odevzdán, Nasazen apod.), dále hodinové ohodnocení (předpokládaná časová náročnost), přiřazené osoby na různých pozicích vůči požadavku (programátor, konzultant, zadavatel apod.) a především interní a externí komunikaci. Systém HelpDesk automaticky rozesílá emaily v případě jakékoli změny (zpráva v komunikaci u požadavku, přidělen na vás nový požadavek, změna stavu požadavku apod.) [2]

2.1.2 Esyco.NET

Produkt pokrývající základní evidenční a ekonomické funkce každé organizace ve sféře obchodu nebo služeb. Systém je optimalizován pro použití ve středně velkých a velkých společnostech s přístupem několika stovek uživatelů. Esyco.NET je rozdělen do několika modulů prostřednictvím kterých jsou pak dále prováděny jednotlivé procesy společnosti. Toto řešení sjednocuje oblasti marketingu, obchodní a výrobní části společnosti, pracovní týmy společnosti, řízení reklamací a servisu a v neposlední řadě i kompletní obchodní procesy od poptávky až po fakturaci, včetně přímé návaznosti na účetnictví a platební styk. [3]

2.1.3 xForLink

Projektová rodina xForLink zahrnuje veškeré produkty společnosti E LINKX a.s. vyznačující se absencí nutnosti produkt instalovat nebo jej fyzicky vlastnit. Jedná se tedy o produkty cloudového řešení, kdy se na veškeré aplikace této rodiny může zákazník přihlásit téměř odkudkoli skrze jeho webový prohlížeč. Do této rodiny spadají například produkty xSklad a xShop.

2.1.4 xSklad

Cloudové řešení pro správu skladu. Uživatelé do systému přistupují přes webové rozhraní v internetovém prohlížeči. Implementuje Warehouse management system (WMS),

který eviduje rozložení zboží na skladě a jeho pohyby. Systém využívá pro rychlou evidenci změn ve skladu mobilní čtečky čárových kódů. [4]

2.1.5 xShop

Moderní e-shop splňující veškeré požadavky moderní doby včetně responzivity. E-shop lze připojit na externí ERP systémy, jako jsou POHODA, Money S3, ABRA a mnoho dalších. [5]

2.2 Pracovní prostory

Individuální odbornou praxi jsem vykonával v ostravském sídle společnosti. Během školení nám společnost E LINKX a.s. poskytovala klidnou kancelář. Zde jsem přibližně třikrát do týdne docházel na školení nebo týmové sezení. Po nástupu do společnosti na pozici stážisty mi bylo přiděleno plnohodnotné pracovní místo mezi ostatními zaměstnanci společnosti. Sídlu firmy jsem tak navštěvoval v pracovních dnech dle mého volného času.

3 Popis použitých technologií

Společnost E LINKX a.s. je dlouhodobým obchodním partnerem společnosti Microsoft. Důkazem tohoto partnerství jsou obdržena ocenění Gold Hosting, Silver OEM a Silver Server Platform, která společnost několikrát obhájila. Není tedy divu, že je společnost E LINKX a.s. úzce spjata s poskytovanými technologiemi a řešeními společnosti Microsoft.

3.1 C#

Jazyk C# je objektově orientovaný programovací jazyk, který vyvinula společnost Microsoft a je součástí takzvaného **.NET frameworku**. Jazyk byl založen na základech jazyků C++ a Java. Jedná se o jazyk, který je neustále ve vývoji. Verze jazyka je udávána verzi použitého **.NET frameworku**. Společnost Microsoft poskytuje rozsáhlou veřejnou dokumentaci a podporu k tomuto jazyku na webu MSDN [6] [7].

3.2 ASP.NET

ASP.NET je určen pro tvorbu webových aplikací a služeb, rovněž je součástí **.NET frameworku** a je nástupcem technologie **ASP**. Tato technologie ulehčuje programátorům přechod z vývoje aplikací pro Windows do prostředí webů. **ASP.NET** spojuje webové technologie (*HTML*, *JavaScript*) s objektově orientovaným programováním v jazycích C#, VB.NET nebo JSript.NET [8].

3.3 T-SQL

Jedná se o proprietární rozšíření jazyka **SQL** od společností Microsoft a Sybase. Toto rozšíření přináší podmíněné zpracování, sekvenční průchody pomocí kurzorů nebo využití zabudovaných funkcí MSSQL serveru. **T-SQL** umožňuje ukládat procedury a tak znovu využívat kód. Díky procedurám, transakcím a triggerům mohou programátoři přenést některou nezbytnou aplikační logiku na stranu serveru.

4 Školení

Po úspěšném absolvování přijímacího pohovoru jsem byl spolu s dalšími čtyřmi studenty VŠB-TUO fakulty FEI přijat do školicího programu společnosti E LINKX a.s. Tento školicí program zahrnoval jedno až dvě sezení týdně s Ing. Romanem Hrdým a Ing. Radkem Petříkem za účelem vzdělání nebo zadání, konzultování a obhájení úkolu. Plnění úkolů probíhalo individuálně, nebo v týmu. Před začátkem každé implementace projektu jsme museli odevzdat naši předběžnou analýzu zadání spolu s hodinovým ohodnocením.

V průběhu dvouměsíčního školení mi bylo zadáno několik úkolů. Před plněním úkolu jsme byli vždy rozděleni do dvoučlenného nebo tříčlenného týmů. V průběhu školení nám byly zadávány i dočasné malé úkoly, které se dále plnily samostatně. Tyto drobné úkoly v této práci podrobněji nerozvádím. Jednalo se však o úkoly výzkumného charakteru.

4.1 Replikace dat

Během prvních dnů školení jsme byli rozděleni do dvou týmů. Byl nám zadán projekt, který byl zprvu charakterizován velice stručně, a to jako aplikace, která bude hlídat soubory nebo složky a replikovat změny na nastavené adresy. Tento projekt musí být implementován pomocí jazyku C# a měl by nalézt uplatnění na firemních serverech. Tento projekt zaměstnancům zjednoduší a zefektivní replikování konfigurací různých webů a dalších služeb. Postupem následujících dnů byly požadavky dále specifikovány a rozšířeny o nové funkcionality. Tyto požadavky si dovoluji podrobněji shrnout v patřičných částech popisu tohoto projektu. Nicméně pár klíčových požadavků uvedu již nyní.

- **Hlídání změn:** Řešení musí automaticky rozpoznat změnu v nastaveném souboru nebo složce. Řešení dále musí umět hlídat více souborů nebo složek na různých místech souborového systému.
- **Replikace změn:** Řešení musí umět změny v souboru či složce přenést i do cílových adres (místa v souborovém systému, kam se mají změny replikovat), dále musí povolit mapování více cílových adres pro dané sledování.
- **Součást systému:** Řešení musí běžet po spuštění operačního systému a až na detekované změny v hlídaných souborech by nemělo uživatele jakkoli rušit.
- **Uživatelské rozhraní:** Řešení musí obsahovat přehledné uživatelské rozhraní, pomocí kterého bude uživatel moci jednoduše a přehledně nastavit sledování souborů a složek spolu s jejich místy cílové replikace.
- **Interakce s uživatelem:** Řešení musí být navrženo tak, aby uživatel mohl replikaci provedených změn zrušit nebo odložit.

Spolu se samotným řešením zadání bylo požadováno vypracovat kompletní uživatelskou a programátorskou dokumentaci pro obsluhu a případnou údržbu tohoto řešení. V neposlední řadě jsme měli vytvořit i prezentaci produktu pro zákazníka. V rámci týmové

práce jsme museli vyhotovit analýzu a svou budoucí práci nad tímto projektem časově nacenit. Celé období řešení tohoto projektu bylo vedeno stylem smyšleného vztahu mezi zákazníkem (Ing. Roman Hrdý a Ing. Radek Petřík) a společností (náš tým).

4.1.1 Tým, analýza částí a nacenění práce

Řešení tohoto projektu jsem vypracoval s Tomášem Bauerem, se kterým jsme v rámci prvního projektu utvořili tým. Ing. Radkem Petříkem a Ing. Romanem Hrdým jsem byl stanoven jako vedoucí našeho týmu a dostal jsem tak v týmu hlavní slovo. Po týmové diskusi jsem rozhodl o rozdělení celkového řešení mezi tři zdánlivě samostatné aplikace. Jednotlivé části finálního řešení jsou Windows služba, konfigurační rozhraní a notifikační aplikace. Každá jedna z těchto tří částí tak bude zajišťovat potřebnou funkcionalitu, na kterou byl zákazníkem kladen důraz.

Poznámka 4.1 Naše první hodinové ohodnocení bylo na 60 hodin práce. Toto ohodnocení se brzy ukázalo jako příliš optimistické a ve finále jsme nad projektem replikace dat strávili celkem 247 hodin. Opravy dle přání zákazníka, studium a podcenění komplexnosti zadání vedlo k čtyřnásobnému navýšení konečného času. Detailnější přehled těchto hodinových ohodnocení naleznete v přílohách na straně 24.

4.1.2 Windows služba

Dílčí část řešení založena jako Windows service pod frameworkem 4.0 .NET. Tato část je hlavním prvkem celého řešení a obsahuje tak většinu z požadovaných funkcionalit. Z hlavních bodů to jsou následující:

- Součást systému
- Hlídání změn
- Replikace změn

Obsahuje ale i další funkcionality, které byly zákazníkem specifikovány později.

- Historie: Služba bude před každou replikací změn evidovat aktuální změny do historie
- Logování: Služba bude své veškeré vnitřní stavy o zachycených a replikovaných změnách zapisovat do Windows logu.
- Email: Služba bude odesílat informativní email v případě chyb při replikaci.
- Hlídání cílových verzí: Služba bude před replikací kontrolovat, zda nebyl cílový soubor lokálně modifikován. (Porovnání zda je soubor na cíli stejný, jako byl soubor na zdroji před provedenou změnou).

4.1.3 Konfigurační aplikace

Část projektu cílena na uživatele a jeho interakci se službou. Obsahuje přehledné grafické uživatelské rozhraní pro nastavení služby. Z důvodů práce s Windows službou potřebuje tato aplikace pro spuštění práva administrátora. Zahrnuje tento hlavní bod:

- Uživatelské rozhraní

4.1.4 Notifikační aplikace

Třetí a poslední část finálního řešení. Skromná aplikace s grafickým rozhraním běžící částečně na pozadí se schopností upozorňovat uživatele na dění pomocí „pop-up“ oken ve stavové části obrazovky operačního systému Windows. V případě změn umožňuje přímou interakci s uživatelem. Je zde tedy zahrnut bod:

- Interakce s uživatelem

4.1.5 Implementace Windows služby

Stěžejní část tohoto projektu implementována v frameworku 4.0 .NET. Pomocí knihovny *System.IO* jsme pracovali se souborovým systémem operačního systému. Klíčovou třídou v tomto projektu byla třída **FileSystemWatcher**, která nám umožňovala jednoduše a efektivně hlídat soubor nebo složku a detekovat tak provedené změny. Jelikož služba neobsahuje žádné grafické rozhraní, byli jsme nuceni vytvořit konfigurační XML soubor. Pomocí tohoto XML souboru službu nastavíme a předáme jí veškeré nutné informace. Pro sledování vnitřních stavů a výpis chyb jsme použili třídu **EventLog** z knihovny *System.Diagnostics*.

Samotná replikace dat nebyla nijak zvláštní výzvou. Z důvodu optimalizace a časového kopírování jednoho souboru na více cílových adres jsme využili třídu **MemoryStream** z již zmiňované knihovny *System.IO*. Při implementaci rozesílání e-mailů a logování vnitřních stavů jsme využili třídy z knihoven *System.Net.Mail* a *System.Diagnostics*.

Praktické využití třídy **FileSystemWatcher** můžete zhlédnout na straně 13, výpis kódu č. 1, kde pomocí *GetEntryAssembly().Location* získám umístění spouštěného kódu (exe soubor služby) a za pomoci statické třídy **Path** a jejích metod tuto adresu upravím na adresu konfiguračního souboru obsahující nastavení. Instance třídy **FileSystemWatcher** vyvolá událost *Changed* v případech, kdy budou velikost nebo čas poslední úpravy sledovaného souboru změněny.

```

private void SetWatcherToConfig()
{
    string finalPath = Path.GetDirectoryName(System.Reflection.Assembly.GetEntryAssembly().
        Location);
    configWatcher = new FileSystemWatcher(finalPath);
    configWatcher.Filter = Path.GetFileName(Path.Combine(finalPath, "watched.config"));
    configWatcher.NotifyFilter = NotifyFilters.LastWrite | NotifyFilters.Size;
    configWatcher.EnableRaisingEvents = true;
    configWatcher.Changed += new FileSystemEventHandler(ConfigChanged);
}

```

Výpis 1: Služba: Hlídní změn nad svým XML souborem

4.1.6 Implementace konfigurační aplikace

Druhá část projektu obsahuje grafické uživatelské rozhraní ulehčující konfiguraci Windows služby. Tato část nemusí být nutně na počítači a není přímo potřebnou částí celého projektu, ale bezesporu ulehčuje uživateli celkovou správu služby. Umožňuje jednoduše konfigurovat adresy skrze dialogy a zamezí možnému zanesení chyb do konfiguračního souboru při ruční editaci.

Uživatelské rozhraní jsem se pokusil navrhnout co nejvíce minimalisticky a přehledně. Proto jsem například zvolil kontejner **TabControl** na rozdíl od standardního **MenuStrip**. Finální podobu uživatelského rozhraní si můžete prohlédnout na obrázku č. 1.

Stěžejní část této aplikace byla správa stávajícího konfiguračního souboru a následně i zapsání nové konfigurace do XML. Z důvodu potřeby jednotné práce se souborem XML jsme založili knihovnu *ConfigurationAccessLibrary*, která tyto úkony zaobaluje a je následně importována i do projektu Windows služby.

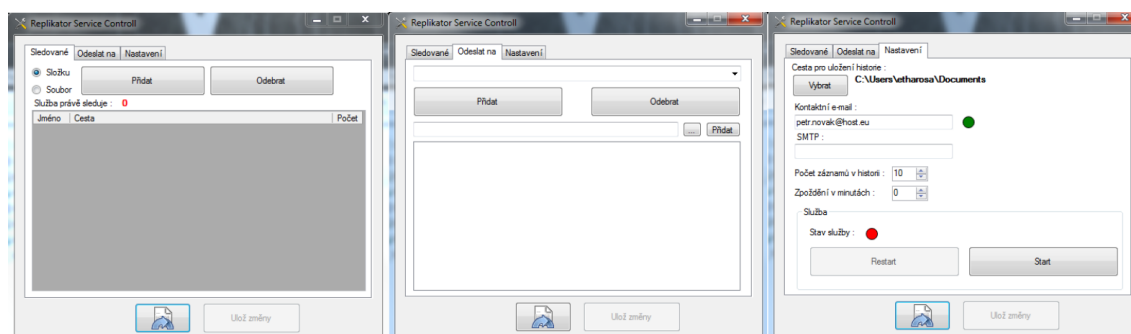
Restrikci funkčnosti pouze v případě existence replikační služby v počítači můžete vidět ve výpisu kódu č. 2. V tomto úryvku lze vidět jak pomocí třídy **ManagementClass** získám všechny nainstalované služby, které následně procházím cyklem **foreach**. Uvnitř každé iterace cyklu zjistím název služby a v případě shody zjistím i umístění. Konfigurační aplikace se v případě nenalezení souboru *watched.config* ukončí, toto ukončení ale už není součástí ukázky kódu. Jedná se pouze o jednoduchou podmínku s použitím metody *File.Exists(string path)*, která by následovala po zmiňovaném bloku kódu.

```

ManagementClass mc = new ManagementClass("Win32_Service");
foreach (ManagementObject mo in mc.GetInstances())
{
    if (mo.GetPropertyValue("Name").ToString() == "ReplikatorService")
    {
        servicePath = mo.GetPropertyValue("PathName").ToString().Trim("");
        servicePath += ".config";
        servicePath = Path.Combine(new FileInfo(servicePath).DirectoryName, "watched.config");
        break;
    }
}

```

Výpis 2: Ověření existence služby a získání její konfigurace



Obrázek 1: GUI: Replikator

4.1.7 Implementace Notifikační aplikace

Třetí a tedy poslední částí tohoto projektu byla notifikační aplikace. Tato aplikace je spuštěná po startu operačního systému a přetrvává na pozadí s notifikační ikonou v system tray. Aplikace se skrze komunikační kanál WCF pravidelně dotazuje Windows službu (Knihovny *System.ServiceProcess* a *System.ServiceModel*), zda byly provedeny změny a zda je potřeba s uživatelem interagovat nebo mu něco sdělit.

Zprávy uživateli zobrazujeme pomocí pop-up oken (*NotifyIcon.ShowBalloonTip*). V případě detekované změny je uživateli zobrazeno drobné okno (*Windows.Forms*), v rámci kterého může uživatel na tuto změnu zareagovat. Uvnitř notifikační aplikace jsou nastaveny dva časovače (*Windows.Forms.Timer*), které řídí dotazování na Windows službu.

4.1.8 Rozšíření: Správa historie

Po navržení a zprovoznění konfigurační aplikace (a tím i konfiguračního souboru) měla služba k dispozici potřebné informace pro správu historie. Pro tyto účely jsem vytvořil knihovnu *HistoryManagement*, která zapouzdřuje veškerou potřebnou práci s historií jako je například hlídání velikosti historie, navrácení poslední známe verze souboru z historie a zaevidování nového souboru do historie. Po sepsání a spuštění této třídy se dosavadní politika replikace změnila z přímé replikace originálu na replikaci kopie z historie.

4.1.9 Meziaplikační komunikace

Pro komunikaci mezi Windows službou a notifikační aplikací jsme využili technologii WCF named pipes, kdy na straně Windows služby je nakonfigurováno rozhraní, skrze které poskytuje předem definované metody a data. Windows služba neví nic o svém okolí, které se jí může skrze toto rozhraní dotazovat. Při spuštění služby dochází pouze k spuštění vlákna, které je připraveno vyřizovat požadavky. Notifikační aplikace má připojenou referenci na toto zveřejněné rozhraní a tím přebírá i její nastavení a dostupné metody.

Na straně Windows služby je vytvořen buffer zpráv, který se neustále plní a je udržována životnost zprávy kolem jedné a půl vteřiny. Zprávy starší jak vteřina a půl jsou z bufferu mazány. Každá zpráva v bufferu obsahuje typ zprávy, text zprávy a ID zprávy.

Na straně notificační aplikace se v případě úspěšného navázání spojení s Windows službou spustí hlavní časovač, který způsobí dotazování každou jednu vteřinu metodou *DoYouHaveSomething()*. Výsledkem takto volané metody je seznam zpráv přijatých z bufferu služby, které jsou dále notificační aplikací vyfiltrovány (dle ID) a zpracovány (příslušně zobrazeny). V opačném případě je notificační aplikace spuštěna v režimu "standby" a pokouší se v pětivteřinovém intervalu navázat spojení s Windows službou.

V průběhu řešení této mezipřikládky komunikace jsme čelili velkému problému v časových odchylkách časovačů. V brzkých stádiích této komunikace se nám tak často stávalo, že se notificační aplikace stihla zeptat metodou *DoYouHaveSomething()* dvakrát v jedné vteřině, nebo naopak ani jednou. V obou případech docházelo k nežádoucímu chování, kdy byla zpráva uživateli zobrazena dvakrát nebo nebyla zobrazena vůbec.

V průběhu podrobnějšího průzkumu tohoto problému jsme změřili maxima odchylek časovačů od nastavené hodnoty, které činily zhruba +200 ms a -50 ms v provozu s rychlostí sepínání časovače každou jednu vteřinu. Trvanlivost zpráv v bufferu na straně služby byla nastavena na vteřinu a půl. Pro eliminování duplicitního zobrazování přijatých zpráv jsme zprávám přidělili jedinečné ID, které je každou zprávou inkrementováno o jedničku. Notificační aplikaci poté stačilo si pouze pamatovat ID naposledy zobrazené zprávy a při příštím čtení zpráv brát v potaz jen ty s větším ID.

4.2 Webové rozhraní pro replikaci dat

Úkolem druhého projektu bylo vytvoření webového rozhraní, které umožní monitorování běhu prvního projektu skrze webový portál. Webový portál musí být implementován pomocí technologie ASP.NET a práce s ním musí být umožněna pouze verifikovaným uživatelům (zaměstnancům). Mezi hlavní požadavky patří možnost sledovat výpis stavů (Logu) v reálném čase spolu s možnostmi tento výpis filtrovat. Dále má portál poskytnout alespoň částečnou schopnost nastavovat sledované a cílové cesty. Dodatečným úkolem pak byla například implementace vícejazyčného rozhraní (lokalizace) a uživatelských rolí (restrikce pravomocí v závislosti na typu přihlášeného uživatele).

4.2.1 Tým a analýza zadání

Na tomto zadání jsem spolupracoval s Lukášem Vrbou, který byl v minulém projektu členem druhého týmu. Finální řešení tohoto projektu mělo spolupracovat s předešlým projektem mého týmu ve složení s Tomášem Bauerem, proto jsem musel Lukáše Vrba seznámit s naším řešením a jeho možnostmi. Během analýzy zadání jsme společně došli k závěru nutnosti toto řešení rozdělit na dvě části.

První část je webová služba zprostředkovávající API pro získání dat z EventLogu systému Windows a dále pro čtení a zápis dat do konfiguračního souboru replikační služby. Tato webová služba bude nainstalována na totožném počítači nebo serveru, kde bude nainstalována i replikační služba (např. interní datový server).

Druhou částí bude webový projekt napsaný v ASP.NET a běžící na IIS někde na veřejném webovém serveru společnosti. Tato část bude získaná data skrze API vizualizovat a interagovat s uživatelem skrze webovou stránku. Práce s webem je podmíněna autorizací uživatele.

4.2.2 Implementace aplikačního rozhraní

První privátní částí tohoto projektu je webová služba poskytující přístupové rozhraní pro získávání dat. Projekt využívá obdobné metody přístupu, jako jsem použil v projektu replikace dat. Na vyžádání čtu z EventLogu operačního systému a vyhledávám zprávy zapsané replikační službou (ukázka níže), čtu a zapisuji konfigurační soubor (opět připojená knihovna *ConfigurationAccessLibrary* z předešlého projektu) a v neposlední řadě ověřuji zadané adresy sledovaných a cílových cest.

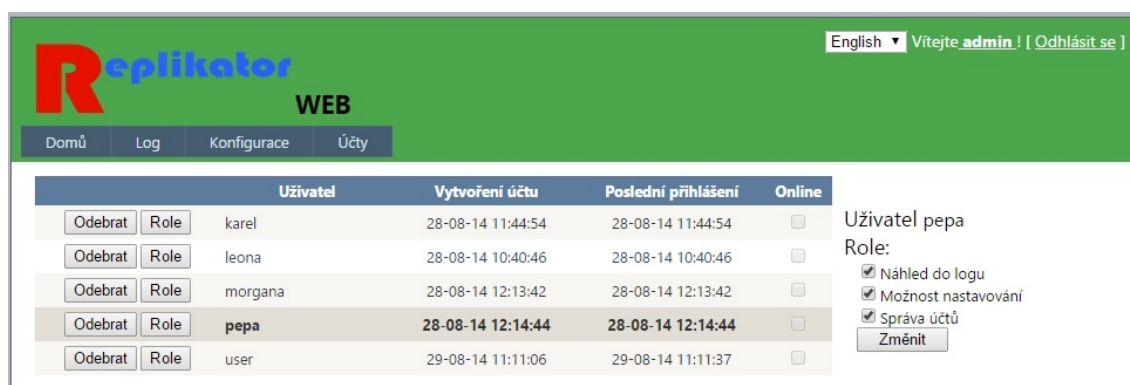
```
[WebMethod]
public List<EventLogEntry> GetEvents()
{
    List<EventLogEntry> tmpList = new List<EventLogEntry>();
    EventLog myFilter = new EventLog();
    myFilter.Source = "Application";
    foreach (EventLogEntry item in myFilter.Entries)
    {
        if (item.TimeGenerated < lowLimit || item.TimeGenerated > topLimit)
            continue;
        if (item.Source != "ReplikatorService")
            continue;
        tmpList.Add(item);
    }
}
```

Výpis 3: Vyhledávání zpráv v EventLogu

4.2.3 Implementace webu

Druhá část projektu je webová stránka založena na technologii ASP.NET, která poběží na veřejném webovém serveru. V projektu jsou použity již dodávané komponenty pro přihlášení, zprávu uživatelů a registraci. Vytvořili jsme jednotlivé stránky pro prohlížení EventLogu vygenerovaných replikační službou a dále stránky pro nastavování služby spolu s nastavením hlídaných a cílových cest.

V rámci dalších vylepšení jsme implementovali lokalizaci i do anglického jazyka. Jazyk uživatele jsme měli uložen v cookies, kam se uložil po jeho volbě. Nastavení zdrojového souboru pro získání textů můžete vidět níže. V projektu byly vytvořeny pro každou stránku (aspx) dva zdrojové soubory, tzn. resource soubory, např. index.aspx.resx a index.aspx.en-gb.resx obsahující překlady obsažených textů.



Obrázek 2: GUI: Web

```
protected override void InitializeCulture ()
{
    if (Request.Cookies["UserSettings"] != null && Request.Cookies["UserSettings"]["Language"] != null)
    {
        string userSettings = Request.Cookies["UserSettings"]["Language"];
        UICulture = userSettings;
        Culture = userSettings;
    }
    base.InitializeCulture ();
}
```

Výpis 4: Přepnutí jazyka zobrazované stránky

4.3 Parametrizace XML

Práce nad projekty byla zpestřena a na jedno odpoledne přerušena zadanou prací pro Ing. R. Petříka. Náplní tohoto zadání bylo porovnávání zdrojových (původních) XML konfiguračních souborů s finálními (ručně modifikovanými) XML konfiguračními soubory. Rozdíly mezi těmito soubory jsme měli nahradit parametrem (proměnou) a tyto parametry nakonec sloučit do již hotového PowerShell skriptu, který tuto ruční modifikaci automatizuje. Zadání obsahovalo zhruba čtyřicet XML souborů.

4.3.1 Použitý software a rozdělení práce

Na tuto práci jsem se přihlásil spolu s dalšími dvěma studenty. Ing. R. Petřík stanovil Tomáše Bauera jako vedoucího pro tuto práci. T. Bauer mezi nás rozdělil rozsahy XML souborů pro porovnání a seznámil nás s formátem zapisování parametrů pro PowerShell script. S porovnáváním rozdílů mezi XML soubory nám pomohl freeware software WinMerge.

4.3.2 Průběh práce a její dokončení

Mým úkolem bylo tedy vzít a porovnat dva XML konfigurační soubory, kde jeden byl aktuální kopií originálu, který je nyní použit, a druhý soubor byl ručně modifikován pro potřeby nového nastavení webů nebo služeb. Často se jednalo o hodnoty elementů, jako jsou například www adresy umístění jednotlivých webů nebo služeb, jejich verze, odkazy na další části konfigurací nebo výčty podporovaných lokalizací, uživatelských účtů a jejich práv. Mnohdy bylo nutné zdrojové XML restrukturalizovat, tedy upravit strukturu XML do nového sledu elementů (některé bloky odebrat nebo přesunout).

Po dokončení parametrizace všech přidělených XML souborů jsem T. Bauerovi zaslal seznam všech svých parametrů a všechny již parametrizované soubory. Společnými silami jsme poté redukovali počet parametrů na základě stejných typů údajů, které tyto parametry zastupovaly a nahrazovaly.

5 Stáž

Počátkem září roku 2014 jsem spolu s dalšími dvěma studenty nastoupil do společnosti E LINKX a.s. jako stážista. Bylo mi přiděleno plnohodnotné pracovní místo mezi ostatními zaměstnanci firmy. Z důvodu ochrany firemního tajemství a citlivosti informací si dovolím následující odstavce této práce pojmut jako stručný popis vykonávané práce bez bližších specifikací a detailnějšího popisu řešení.

5.1 Oddělení FLEXICA

Po nástupu do společnosti na pozici stážisty jsem byl přiřazen do oddělení takzvaného FLEXICA týmu. Zde jsem se v prvních dnech seznámil s provozem společnosti, se zaměstnaneckým systémem HelpDesk a firemní infrastrukturou.

Toto oddělení je zaměřeno na implementaci cloudových řešení a dalších aplikací bez nutnosti instalace, tedy s kompletním přístupem a správou skrze internetový prohlížeč.

5.1.1 Testování a testovací scénáře

Mým úkolem bylo z počátku náhodné testování jejich nového produktu z rodiny xFor-Link a objevené chyby sepsat a následně předat na oddělení Supportu. V následujících dnech se stalo testování řízené a probíhalo na základě testovacích scénářů. Na základě těchto striktně daných scénářů jsem produkt procházel a testoval. Nalezené chyby jsem zanašel zpět do scénáře s dostačujícím popisem pro zpětné vyvolání chyby opravujícím programátorem. Ověřené a doplněné scénáře jsem odevzdal přidělené osobě z oddělení Supportu.

V posledních dnech mého působení na oddělení FLEXICA jsem vypomáhal s návrhem testovacího scénáře v rámci nových částí testované aplikace. Z důvodů neznalosti kompletních návazností testované aplikace se jednalo pouze o hrubý popis či kostru budoucího testovacího scénáře.

5.2 Oddělení Logistiky

Po několikadenním rozkoukání na oddělení FLEXICA jsem byl přiřazen pod Ing. Martina Gürthera. Náplní mé práce byl vývoj nových rozšiřujících částí stávajícího logistického řešení pro jednoho z dlouhodobých klientů společnosti E LINKX a.s.

Mou náplní práce na tomto oddělení bylo:

- Rozšiřování existující databáze o nové tabulky a číselníky
- Tvorba procedur obhospodařující nové tabulky a číselníky v databázi
- Vývoj nových částí GUI pro nové tabulky
- Vývoj dalších manipulací s daty (Exporty, přepočty, grafy apod.)
- Opravy chyb v již stávajících částech programu.

5.2.1 Tvorba tabulek a číselníků

Jeden z prvních úkolů na novém oddělení byla tvorba nových číselníku a tabulek pro právě připravované rozšíření aplikace. Pro každou novou tabulku v databázi bylo zapotřebí vytvořit tři procedury pro práci s ní. Procedury se rozeznávaly podle koncovek v názvu

- PG_NameList - procedura pro navrácení dat, obsahuje klauzuli SELECT
- PG_NameSave - procedura určená pro akce typu INSERT a UPDATE
- PG_NameDelete - procedura obsahující klauzuli UPDATE Reject = 1 (*)

Klient vlastní rozsáhlou mezinárodní databázovou síť. Bylo tedy nutné veškeré mnou provedené změny vložit do replikačních skriptů, které mé změny přenesou na všechny servery při příštím nasazování verze produktu.

Poznámka 5.1 *Oddělení logistiky vede politiku minimálního mazání záznamů, proto má každý řádek v tabulkách sloupec **Reject**, pomocí kterého je v případě nastavení na 1 (true) řádek brán jako neplatný (smazaný).

5.2.2 Vývoj produktu

Po obeznámení s databází a prací s ní jsem vytvářel v produktu pro nové tabulky uživatelské rozhraní. Často se jednalo o formuláře s filtrem a tabulkou (Grid) pro zobrazení filtrovaných dat. V drtivé většině případů byl formulář rozšířen i o možnosti vkládat, upravovat a mazat data. V pozdějším stádiu prací se staly nové formuláře více komplexní co do závislosti dat a funkcionality. Přístup do některých formulářů byl podmíněn typem přihlášeného uživatele nebo místem přihlášení (server).

5.2.3 Generování cenové nabídky

Po dokončení rozšiřování databáze a GUI aplikace mi byl zadán jedinečný úkol generování dokumentů na základě dat v databázi. Nic podobného nebylo v aplikaci doposud vytvořeno. Nad tímto úkolem jsem strávil 130 hodin čistého pracovního času. Nemalá část tohoto času byla strávena při samostudiu knihovny *DocumentFormat.OpenXml* [9]. Strukturu openXML dokumentu si můžete prohlédnout na straně 26.

Zákazník mi dodal českou i slovenskou předlohu nabídek. Nabídky se od sebe navzájem lišily rozsahem generovaných dat a strukturou souborů. Tento rozsah a struktura musí být při generování zachována.

Většina měněných (tedy generovaných) částí byly tabulky, proto jsem vytvořil metodu **WDAddTable**, která vloží tabulku do souboru dle vzoru tabulek od zákazníka (bližší seznámení s metodou na straně 27, výpis kódu č. 7.). Tuto metodu jsem poskytl i T. Bauevovi, který mezitím pracoval na generování smluv.

V průběhu samostudia knihovny *DocumentFormat.OpenXml* jsem přišel na pohodlný způsob nahrazování textů v podobě nahrazování klíčů. Pro snazší a přesnější vyhledávání jsem zvolil skladby klíče v podobě #NazevKlice#. V následující ukázce lze vidět

hloubka nutného zanoření pro získání textu obsaženého v odstavci. Vyhledávaný text musel být v jednotném formátu, v opačném případě by text byl rozdělen mezi více bloků třídy Run a vyhledávání by se tak stalo neúčinné.

```
using (WordprocessingDocument wdDoc = WordprocessingDocument.Open(fileDest, true))
{
    var Paras = wdDoc.MainDocumentPart.Document.Body.Descendants<Paragraph>();
    foreach (var para in Paras)
        foreach (var run in para.Descendants<Run>())
            foreach (var text in run.Descendants<Text>())
            {
                if (text.Text.Contains("#Klic#"))
                    text.Text = "Dnes_je_sobota"
            }
}
```

Výpis 5: Vyhledávání klíče v odstavcích

6 Závěr

6.1 Uplatnění teoretických a praktických znalostí ze studia

Mé dosavadní studium mne dostatečně připravilo na vstupní pohovor u společnosti E LINKX a.s., který jsem absolvoval koncem 4. semestru. Mé dobré obecné znalosti jazyka C# vedly pouze k nutnosti krátkých studií daných problémů, se kterými jsem se prozatím nesetkal (Hlídání změn na disku, logování, mezipřikáční komunikace apod.). Mé dosavadní znalosti technologie .NET a databázových systémů výrazně zjednodušily průběh školícího programu a plnění úkolů během něj.

6.2 Scházející odborné dovednosti a znalosti

V průběhu školení probíhalo sezení s Ing. Radkem Petříkem, který nás seznamoval s různými způsoby řešení daných problémů a objasňoval využití některých pro nás nových tříd jazyka C#. Byl jsem seznámen s tvorbou Windows služeb, použitím EventLogu, s protokolem SOAP a WCF. Prohloubení znalosti v oblastech Thread, ThreadPool a lambda výrazů. Nezanedbatelné jsou i dovednosti z oblasti návrhu softwaru, týmové implementace a komunikace se zákazníkem.

6.3 Shrnutí

Absolvování odborné praxe jakožto téma bakalářské práce jsem si zvolil zejména z důvodu získání dnes tak důležité praxe v oboru, na kterou se budoucí zaměstnavatelé úpěnlivě dívají. S výběrem společnosti mi vypomohla prezentace firem organizovaná školou spolu se seznamem spolupracujících firem v systému KatIs. Společnost E LINKX a.s. mě svou prezentací a zaměřením oslovila nejvíce. Již na vstupním pohovoru jsem byl mile překvapen přátelským přístupem, který přetrval po celou dobu školení i následné stáže. Vedoucí vývojových oddělení, pod kterými jsem v průběhu stáže pracoval, byli velice zkušené a přátelští lidé.

Byl to náročný rok doprovázen neustálým nedostatkem času. Navzdory všemu to byl pro mě velice úspěšný rok, kterého nelituji a doporučil bych individuální odbornou praxi všem budoucím studentům, ne jen těm z vysokých škol. Je to perfektní způsob, jak získat cennou praxi ještě před ukončením studia, a zvýšit tak svou hodnotu na trhu práce. Současně pro společnost je to skvělý způsob, jak si "vychovat" budoucího zaměstnance.

Mé působení v provozu společnosti započalo v září roku 2014 a ke dni 31.3. 2015 jsem ve společnosti odpracoval 470 hodin. Spolu s 208 hodinami strávenými v školícím programu před stáží, popisuje tato práce 678 hodin odborné praxe ve společnosti E LINKX a.s.

Michael Libotovský

7 Reference

- [1] Popis společnosti - *O společnosti* [online].Elink-x a.s. [cit. 2015-03-28]. Dostupné z: <http://www.elinkx.cz/o-spolecnosti>
- [2] HelpDesk - *Technická podpora HelpDesk* [online].Elink-x a.s. [cit. 2015-03-28]. Dostupné z: <http://www.elinkx.cz/helpdesk>
- [3] Esyco.NET - *Informační systém Esyco.NET* [online].Elink-x a.s. [cit. 2015-03-28]. Dostupné z: <http://www.elinkx.cz/esyco-net>
- [4] xSklad - *xSklad - sklad s přehledem* [online].Elink-x a.s. [cit. 2015-03-28]. Dostupné z: <http://www.elinkx.cz/xsklad>
- [5] xShop - *Jeden e-shop pro všechny - xShop.link* [online].Elink-x a.s. [cit. 2015-03-28]. Dostupné z: <http://www.elinkx.cz/xshop>
- [6] C# - *Visual C#* [online].Microsoft®[cit. 2015-03-28]. Dostupné z: <https://msdn.microsoft.com/cs-cz/library/kx37x362.aspx>
- [7] MSDN - *Microsoft Developer Network* [online].Microsoft®[cit. 2015-03-28]. Dostupné z: <https://msdn.microsoft.com/cs-cz/default.aspx>
- [8] ASP.NET - *The ASP.NET Site* [online].Microsoft®[cit. 2015-03-28]. Dostupné z: <http://www.asp.net/>
- [9] OpenXML - *Word processing (Open XML SDK)* [online].Microsoft®[cit. 2015-03-28]. Dostupné z: <https://msdn.microsoft.com/en-us/library/office/cc850833.aspx>

A Hodinové ohodnocení replikace dat

Hodinové ohodnocení pro projekt replikace dat. Tabulky jsou rozděleny do jednotlivých částí tohoto projektu a obsahují jak původně odhadovaný čas, tak i finální strávenou dobu.

Problematika	Předpoklad [h]	Realita [h]
Tvorba GUI	1	11
Mapování adres - Tvorba třídy	1	1
Mapování adres - Propojení s GUI	2	1
Psaní do souboru	4	6
Čtení ze souboru	2	1
Naplnění tříd ze souboru	1	1
Testy s manipulací souboru	1	1
Verifikace dat	1	1
Práce se službou	3	3
Test	2	1
Celkem hodin	18	27

Tabulka 1: Tabulka ohodnocení: Konfigurační aplikace

Problematika	Předpoklad [h]	Realita [h]
Načtení nastavení ze souboru	1	1
Ověření celistvosti	3	6
Hlídání adresářů	5	10
Test prozatimní celistvosti služby	2	2
Reakce na změnu - Odeslat	2	9
Reakce na změnu - Opozdit	3	7
Reakce na změnu - Zrušit	1	6
Implementace fronty změn	1	3
Test reakcí	2	2
Správa Historie	3	13
Tvorba Logu	1	3
Zaslání Emailu	1	1
Nastavení komunikace WCF	2	6
Celkem hodin	27	69

Tabulka 2: Tabulka ohodnocení: Služba

Problematika	Předpoklad [h]	Realita [h]
Nastavení komunikace WCF	2	1
Test základní komunikace	1	1
Tvorba GUI	1	6
Nastavení reakcí	3	25
Nastavení pup-up okna	1	1
Test notifikace a komunikace	2	2
Finální test	5	8
Celkem hodin	15	44

Tabulka 3: Tabulka ohodnocení: Notifikační aplikace

Problematika	Předpoklad [h]	Realita [h]
Korekce kódů, ošetřování chyb a opravy	0	51
Prezentace	0	8
Učení WCF	0	6
Učení config souborů	0	8
Tvorba technické dokumentace	0	18
Tvorba uživatelské dokumentace	0	16
Celkem hodin	0	107

Tabulka 4: Tabulka ohodnocení: Ostatní

Část projektu	Předpoklad [h]	Realita [h]
Konfigurační aplikace	18	27
Služba	27	69
Notifikační aplikace	15	44
Ostatní	0	107
Celkem hodin	60	247

Tabulka 5: Tabulka ohodnocení: Projekt replikace dat

B Ukázka openXML dokumentu

Jedná se o XML s velice členitou strukturou, tato struktura je rovněž přenesená do jazyka C#. Pro každý element v XML existuje třída v knihovně *DocumentFormat.OpenXml*. Pro elementy použité v ukázkovém XML uvedu třídy z knihovny *DocumentFormat.OpenXml*

- `<w:p>` - Paragraph: Soubor je rozdělen do odstavců.
- `<w:pPr>` - ParagraphProperties: Třída zaobalující vlastnosti odstavce.
- `<w:pStyle />` - StyleName / Style: Třída definující styl odstavce dle přednastavených možností nebo uživatelem definovaný styl.
- `<w:spacing />` - SpacingBetweenLines: definuje odsazení odstavce.
- `<w:r>` - Run: Podsekce odstavce, často se člení dle bližšího formátu samostatného textu uvnitř odstavce.
- `<w:rPr>` RunProperties: Třída zaobalující vlastnosti jednotlivého Runu
- `<w:sz />` Size: Udává velikost textu.
- `<w:t>` Text: Třída držící samotný text

```

<w:body>
  <w:p>
    <w:pPr>
      <w:pStyle w:val="Heading1"/>
      <w:spacing w:before="120" w:after="120"/>
    </w:pPr>
    <w:r>
      <w:rPr>
        <w:sz w:val="12"/>
        <w:t>Introduction open</w:t>
      </w:rPr>
    </w:r>
    <w:r>
      <w:rPr>
        <w:sz w:val="18"/>
        <w:t>XML</w:t>
      </w:rPr>
    </w:r>
  </w:p>
  ...

```

Výpis 6: Ukázka dokumentu v openXML

C Popis metody WAddTable

Vstupní parametry metody:

- string fileName - Cesta k souboru, do kterého se má tabulka vložit.
- string MergedFirstLine - První řádek tabulky obsahující její název
- DataTable data - Obsah tabulky v třídě DataTable knihovny *System.Data*.
- string replaceString - Klíč označující místo vložení tabulky.
- List<TableType> TType - List výčtového typu formátu (popis níže).
- int ManualWidth - Vstupní parametr umožňující ovlivnit šířku tabulky.

```
public void WAddTable(string fileName, string MergedFirstLine, DataTable data, string  
replaceString, List<TableType> TType, int ManualWidth)
```

Výpis 7: Signatura metody WAddTable

D Výčet dostupných formátů tabulky

- HeaderTableFullWidth - Nastaví šířku tabulky na 100 % šířky stránky
- None - Vypne jakékoli formátování
- CollHeader - Tabulka bude obsahovat řádek s popisy sloupců.
- RowSpacing - Data tabulky budou odsazena.
- BoldData - Data tabulky budou tučným písmem.
- RightAligment a CenterAligment - Data tabulky budou zarovnána na střed nebo doprava.
- CenterAligmentAll - Vše bude v tabulce zarovnáno na střed.
- GreyText - Data tabulky budou šedým písmem.
- BoldAtLeftCollOnly - Pouze první sloupec je tučným písmem.
- BigInputData - Písma tabulky jsou zmenšena (pro rozsáhle tabulky).
- SK - Přepne paletu používaných barev do schématu pro slovenskou nabídku.
- SpaceAfter - Za vloženou tabulku se vloží i prázdný odstavec.
- NewPageAfter - Do vloženého prázdného odstavce za tabulkou bude vložen i konec stránky (PageBreaker).

```
public enum TableType
{
    HeaderTableFullWidth, None, CollHeader, RowSpacing, BoldData, RightAligment,
    CenterAligment, CenterAligmentAll, GreyText, BoldAtLeftCollOnly, BigInputData,
    SK, SpaceAfter, NewPageAfter
}
```

Výpis 8: Výčet formátů kterými lze ovlivnit finální vzhled tabulky